Vitis-Tutorials/Hardware_Acceleration /Feature_Tutorials /03-dataflow_debug_and_optimization

612k0023c

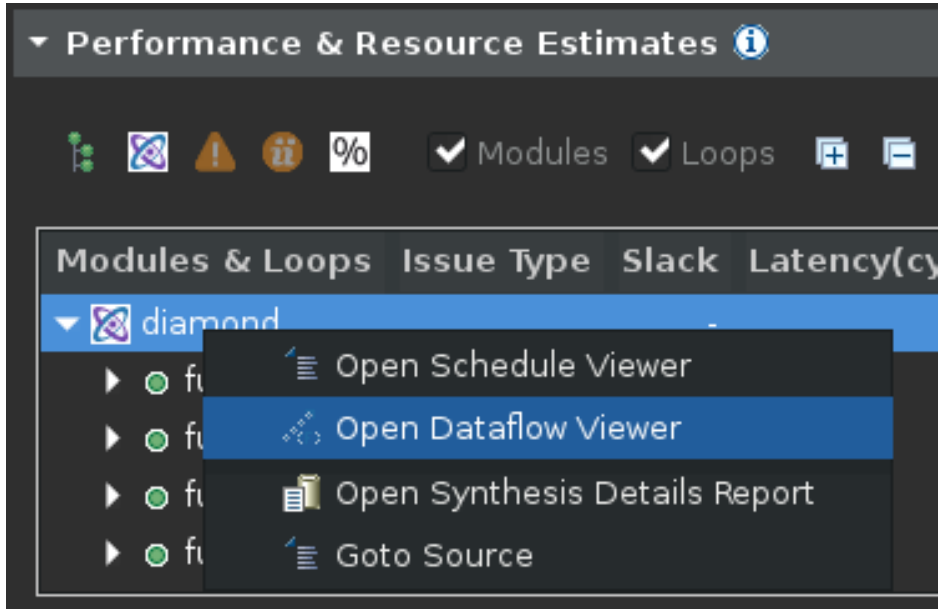
Hao Chen

Outline

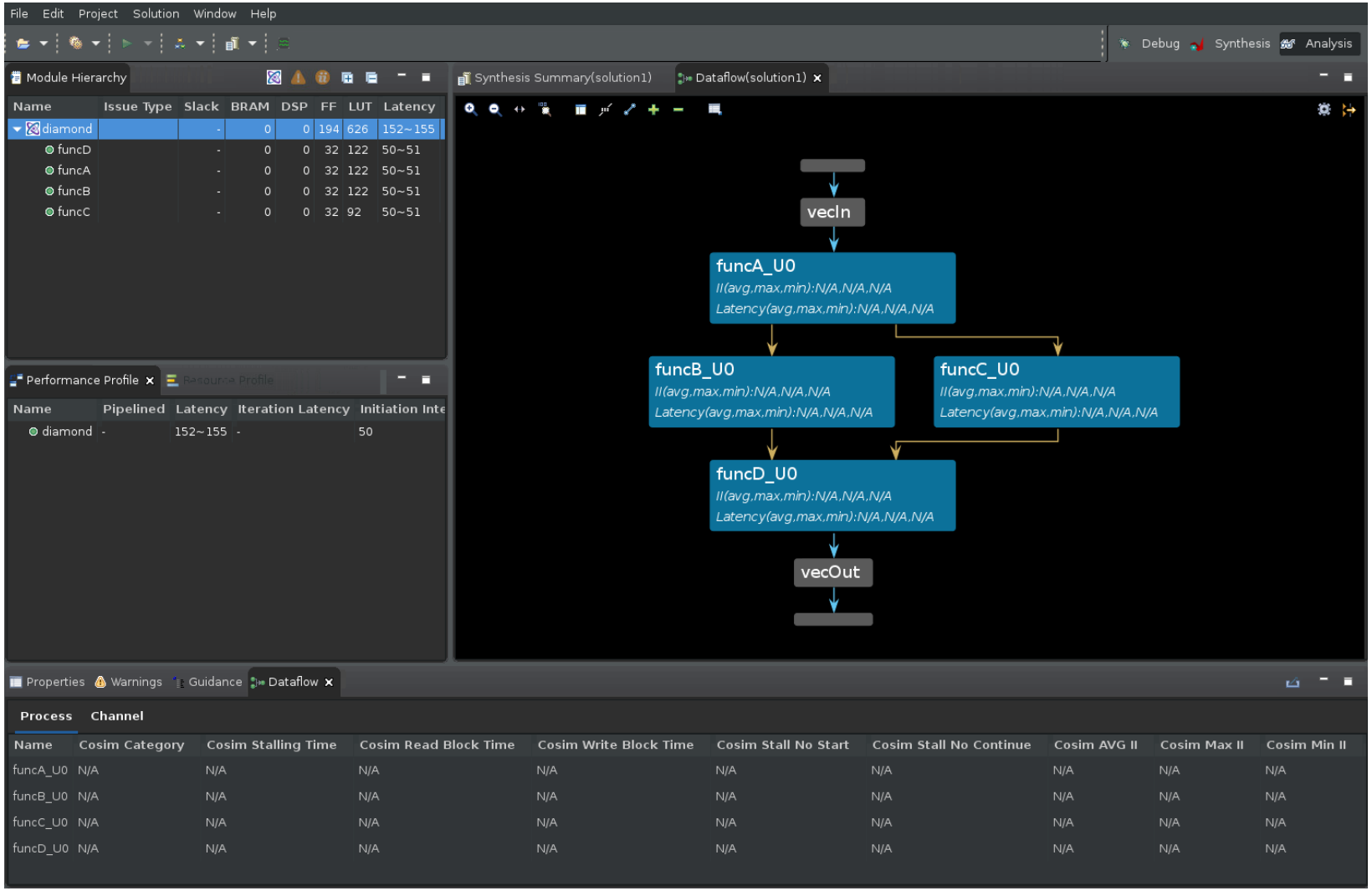
- Dataflow Viewer Basics
- FIFO Sizing and Deadlock

First Lab – Dataflow Viewer basics

- $cd 03-dataflow_debug_and_optimization/reference-files/dataflow
- $vitis_hls -p script.tcl
- **Run C Simulation / C Synthesis**
- **Open Dataflow Viewer:**

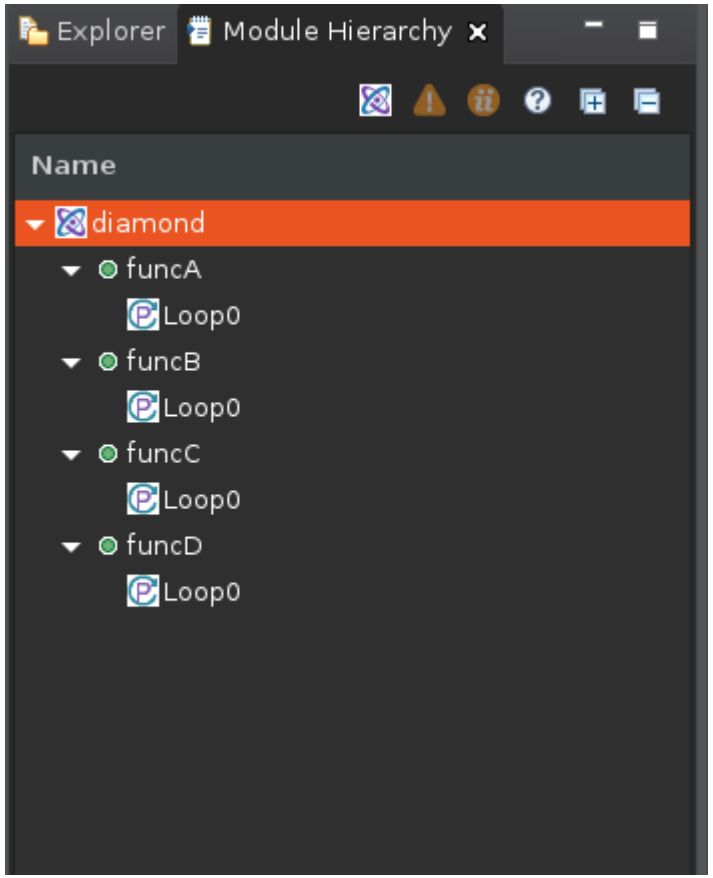


First Lab – Dataflow Viewer basics Cont.

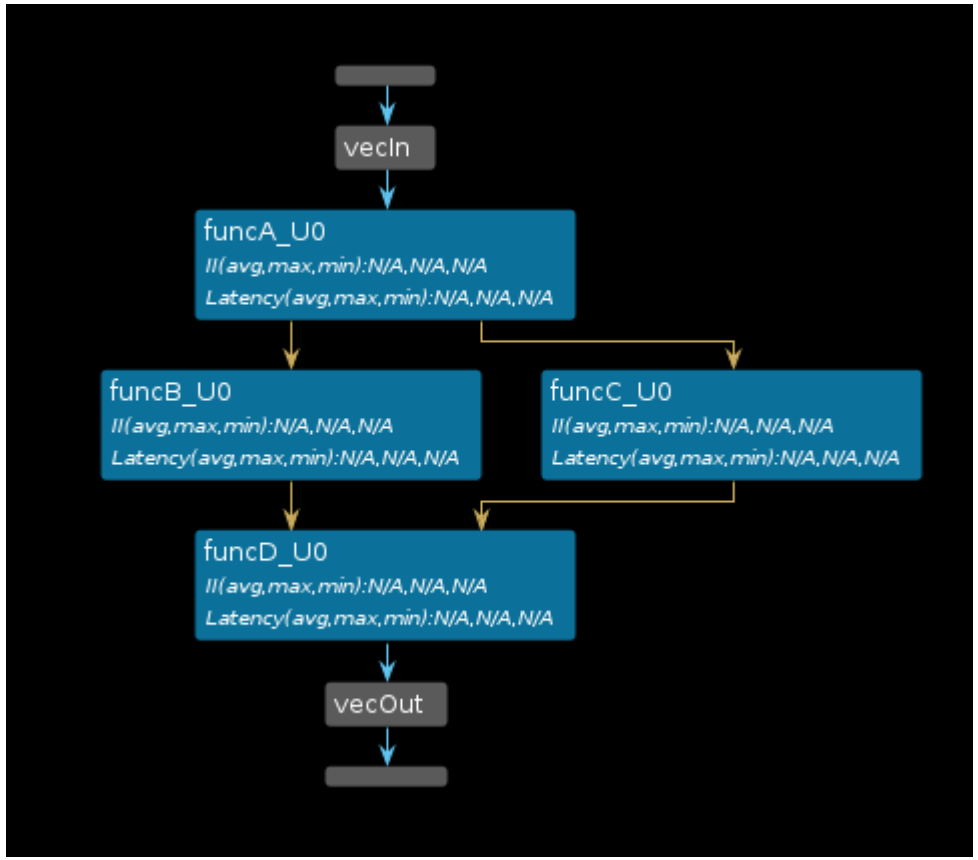


First Lab – Dataflow Viewer basics Cont.

- Module Hierarchy view:

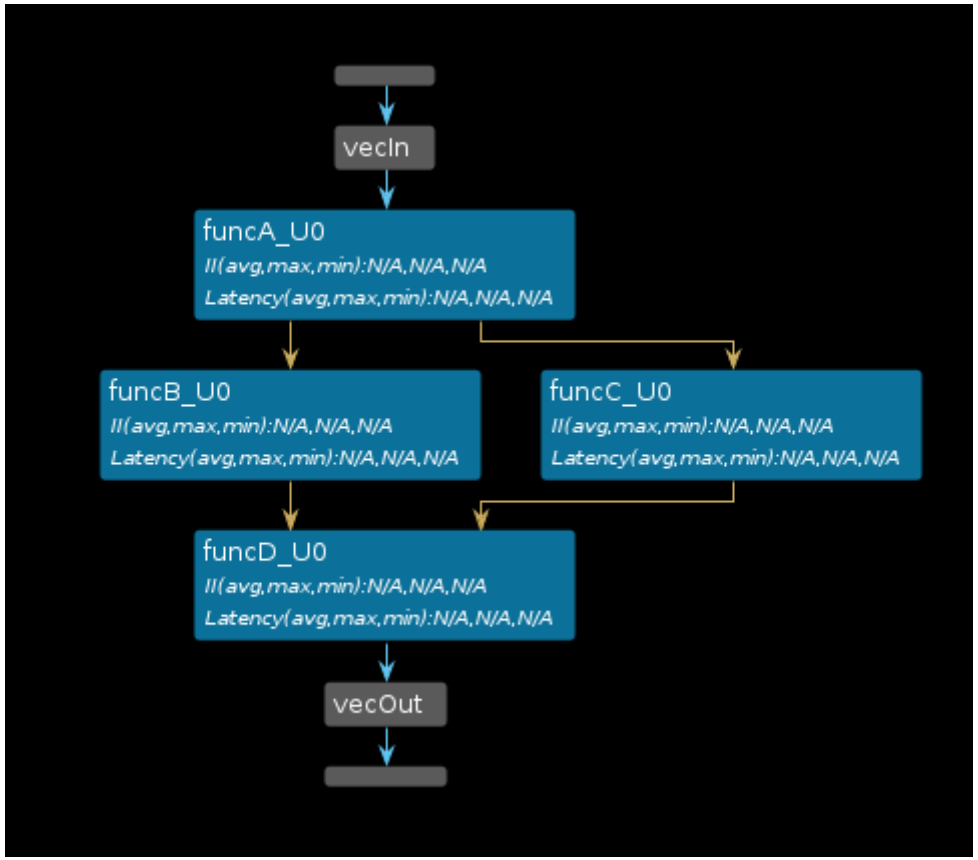


- The Dataflow Graph Pane



First Lab – Dataflow Viewer basics Cont.

- The Dataflow Graph Pane
  - The edges in the graph (shown as blue, green, or gold arrows) represent the communication between the functions:
    - Blue edges represent data dependencies between the functions
    - Green edges represent the inferred FIFO channels between the functions
    - Gold edges represent the inferred PIPO channels



First Lab – Dataflow Viewer basics Cont.

- Dataflow Properties Table
  - At the bottom part of the GUI is the Dataflow Properties table which shows various details about the dataflow processes (or functions) and the dataflow channels.

Console | Errors | Warnings | Guidance | Properties | Man Pages | Git Repositories | Modules/Loops | Dataflow

Process | Channel

Name	Cosim Category	Cosim Stalling Time	FIFO EMPTY	FIFO FULL	Cosim Stall No Start	Cosim Stall No Continue	Cosim AVG II	C
funcA_U0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/
funcB_U0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/
funcC_U0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/
funcD_U0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/

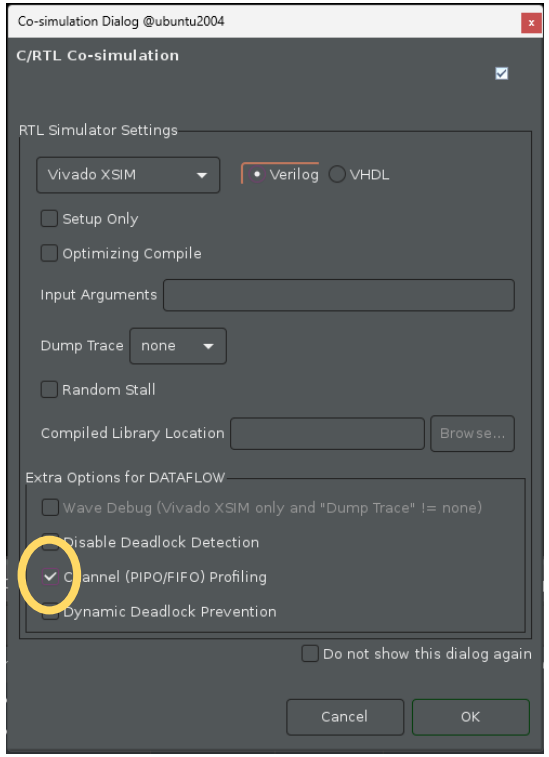
First Lab – Dataflow Viewer basics Cont.

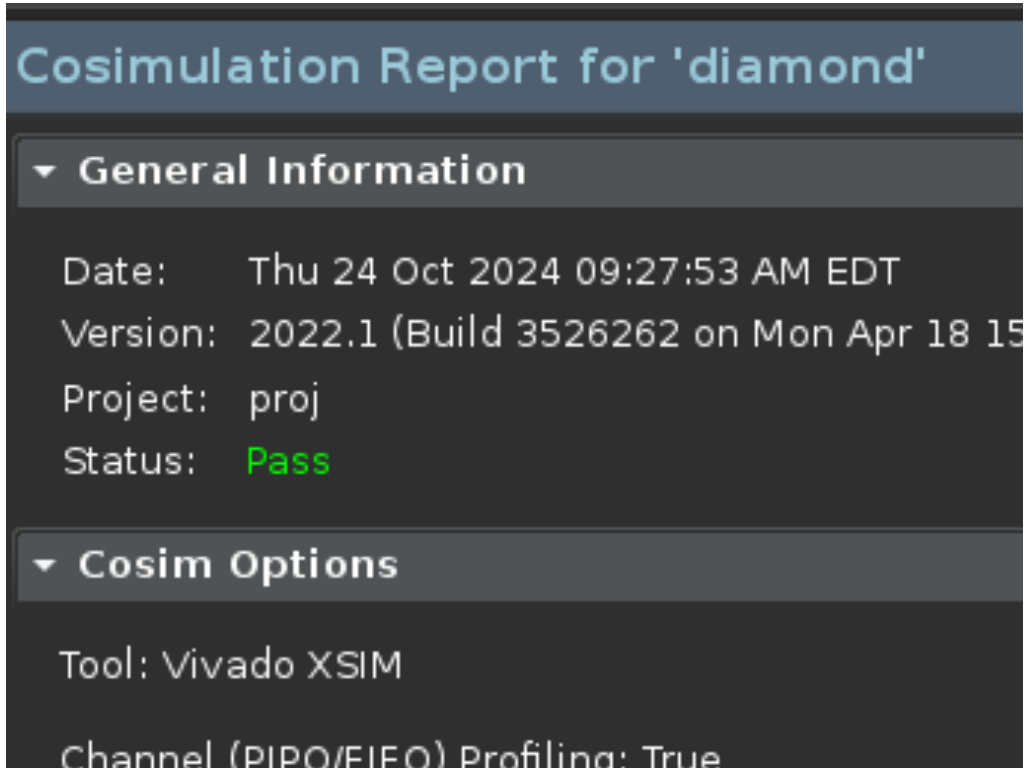
- It is important to understand that the dataflow optimization is a dynamic optimization, unlike pipelining which is a static optimization. Because of this, while the compiler informs you of implementing the dataflow optimization, the effects of the optimization **cannot be seen until after running RTL co-simulation**. The process or channel details are marked as N/A as shown in the figure below until RTL co-simulation has generated performance data.

First Lab – Dataflow Viewer basics Cont.

Viewing the Dataflow Graph after RTL co-simulation

- select the **Solution > Run C/RTL Co-Simulation** command

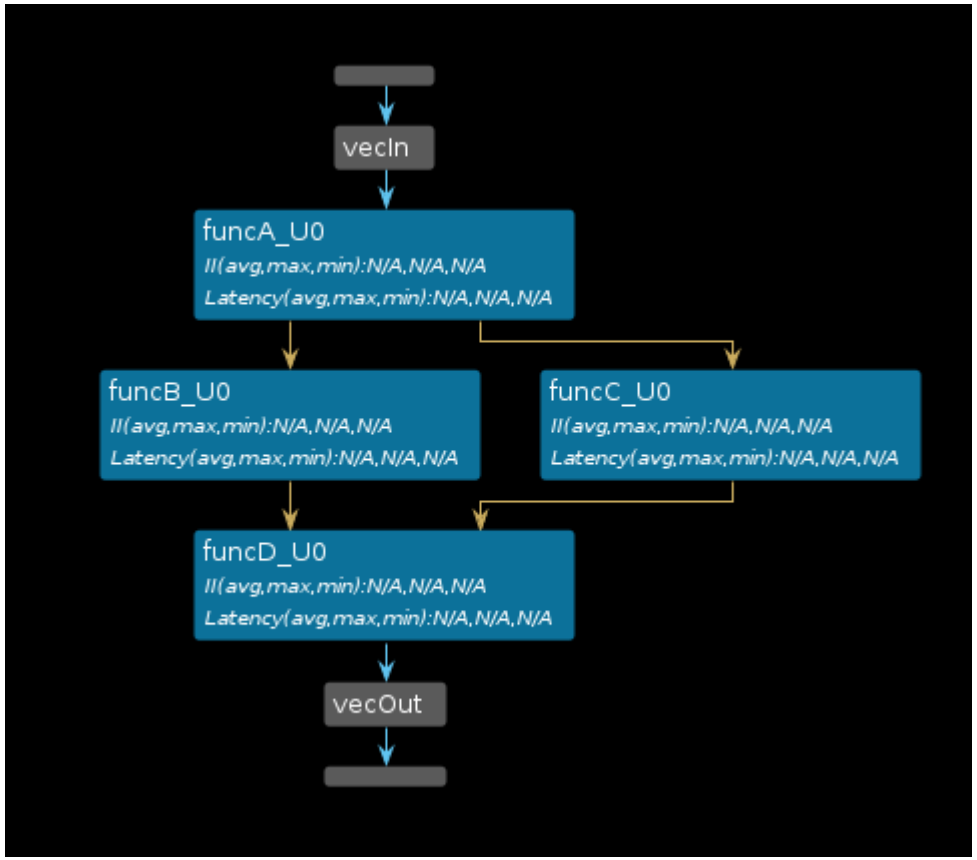


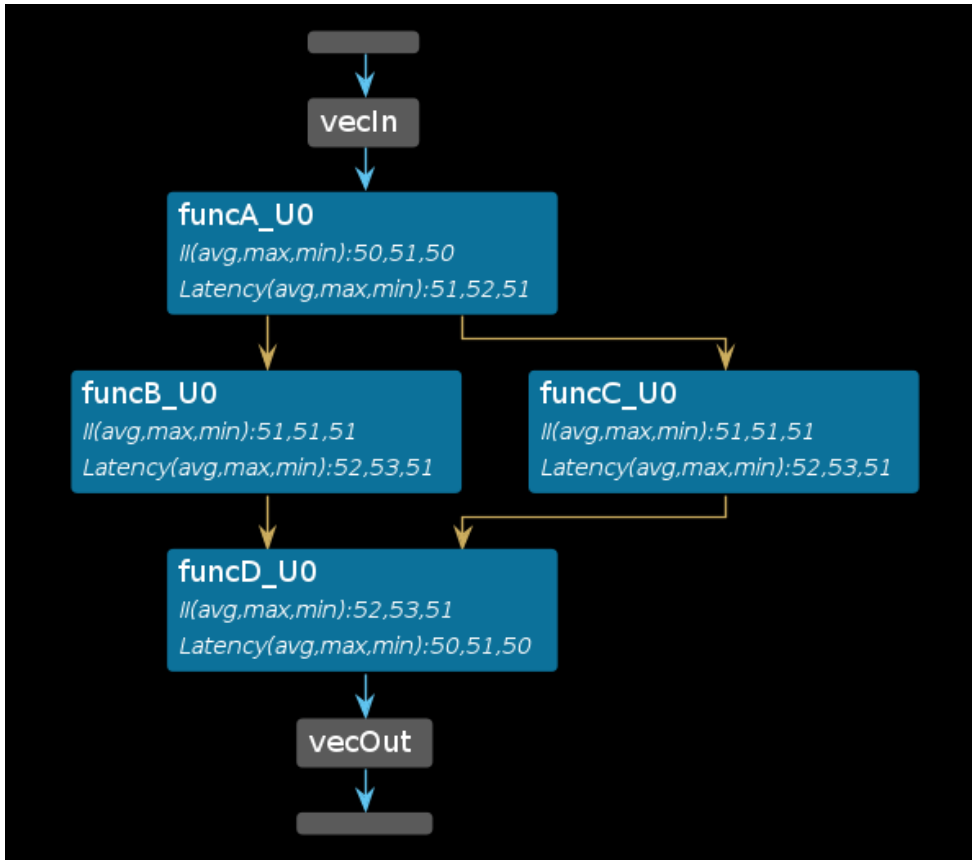


First Lab – Dataflow Viewer basics Cont.

Viewing the Dataflow Graph after RTL co-simulation

- After the co-simulation completes, select **Analysis** in the upper right hand corner of the screen to switch to the Analysis perspective.

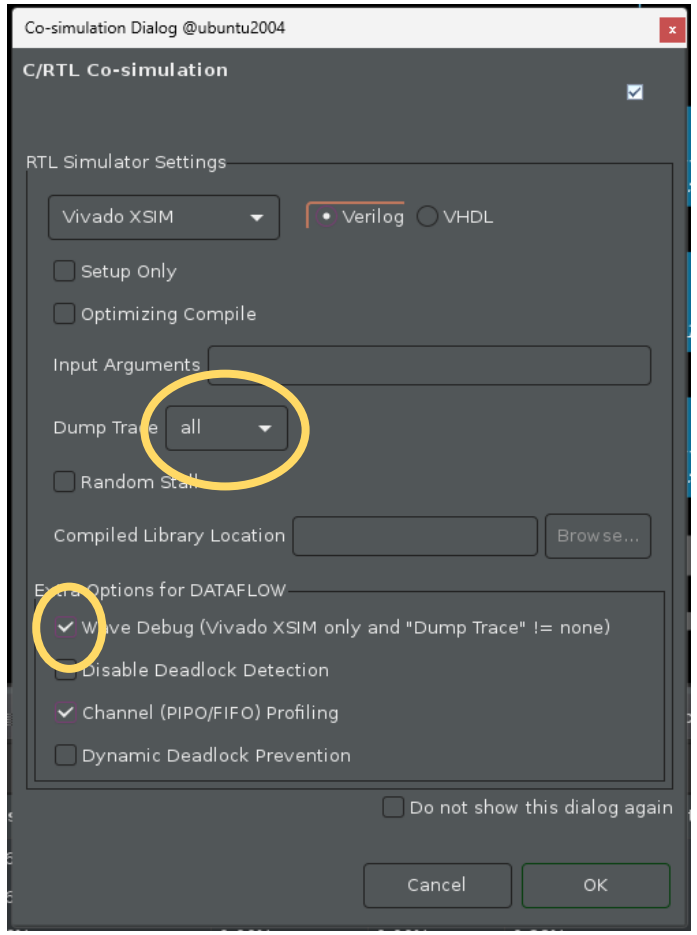




First Lab – Dataflow Viewer basics Cont.

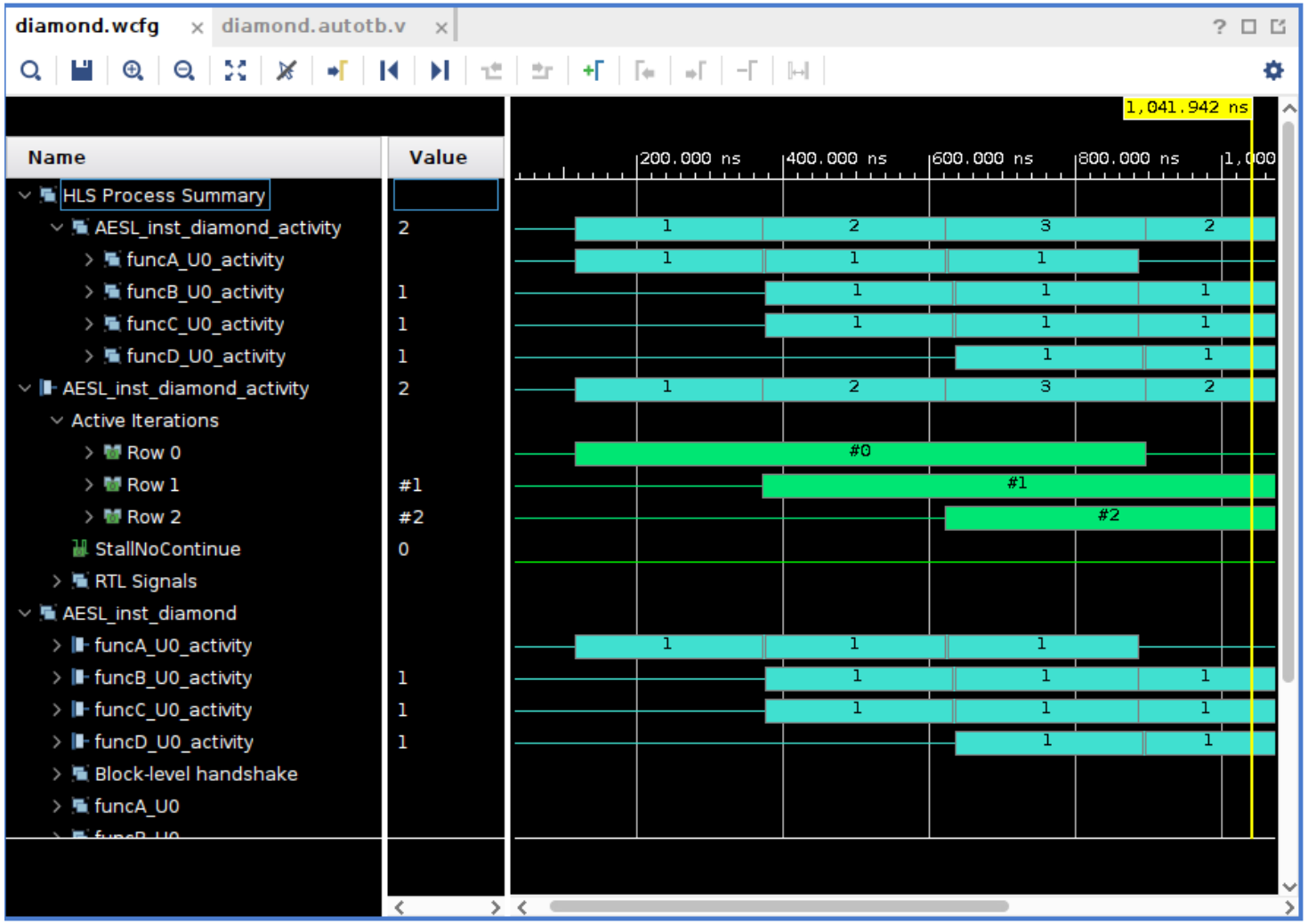
Viewing Dataflow Performance using Waveforms

- Ensure that the **Vivado XSIM** simulator is chosen
- Select **all** for the **Dump Trace** option to trace all ports and signals. Note: This is a small design and so we can dump and trace all the signals. For a large design, this might cause an increased simulation run time as well as the creation of a large waveform database.
- Enable the **Wave Debug** option to interactive launch the XSIM waveform viewer during simulation.
- Enable the **Channel (PIPO/FIFO) Profiling** checkbox.
- Click **OK**.



First Lab – Dataflow Viewer basics Cont.

Viewing Dataflow Performance using Waveforms



Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

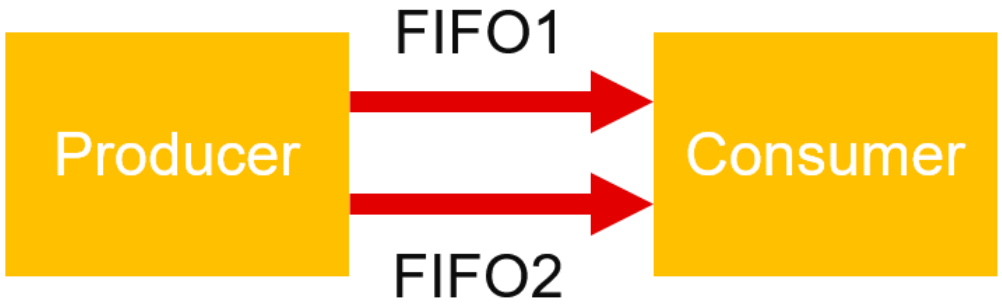
Types of channels

Channel Type	Examples	Created By
FIFO	Streams (including hls::streams and streamed arrays)	User
	Scalar propagation FIFOs	Tool
	Streams of blocks	User

Channel Type	Examples	Created By
PIPO	PIPO	User
	Task Level FIFOs (TLF)	Tool
	Input and output ports to the upper level	User

Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

Deadlock Detection and Analysis



- Case 1:
  - Producer alternately writes to FIFO1, FIFO2, FIFO1, FIFO2, and so on.
  - Consumer alternately reads from FIFO1, FIFO2, FIFO1, FIFO2, and so on.
  - A depth of 1 for both FIFOs is enough to avoid deadlocks (and the default depth of 2 optimizes for performance).
- Case 2 (same structure):
  - Producer writes to FIFO1 for N times, then to FIFO2 for N times
  - Consumer alternately reads from FIFO1, FIFO2, FIFO1, FIFO2, and so on.
  - A depth of N is necessary for FIFO1 (and the default depth of 2 for FIFO2 is optimal for performance)

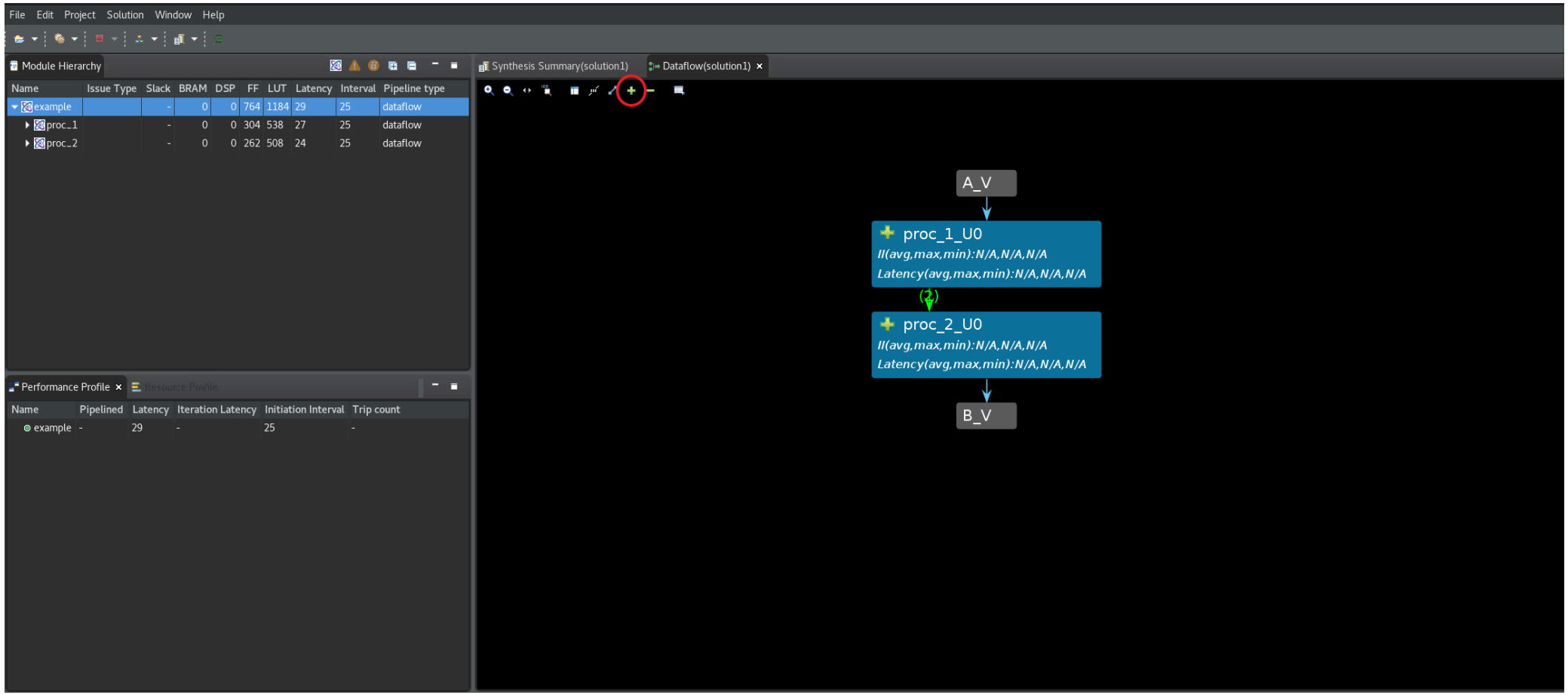
Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

Deadlock Detection and Analysis

- $cd 03-dataflow_debug_and_optimization/reference-files/deadlock
- $vitis_hls -p script.tcl
- Run C-sim, C-synthesis, Co-sim

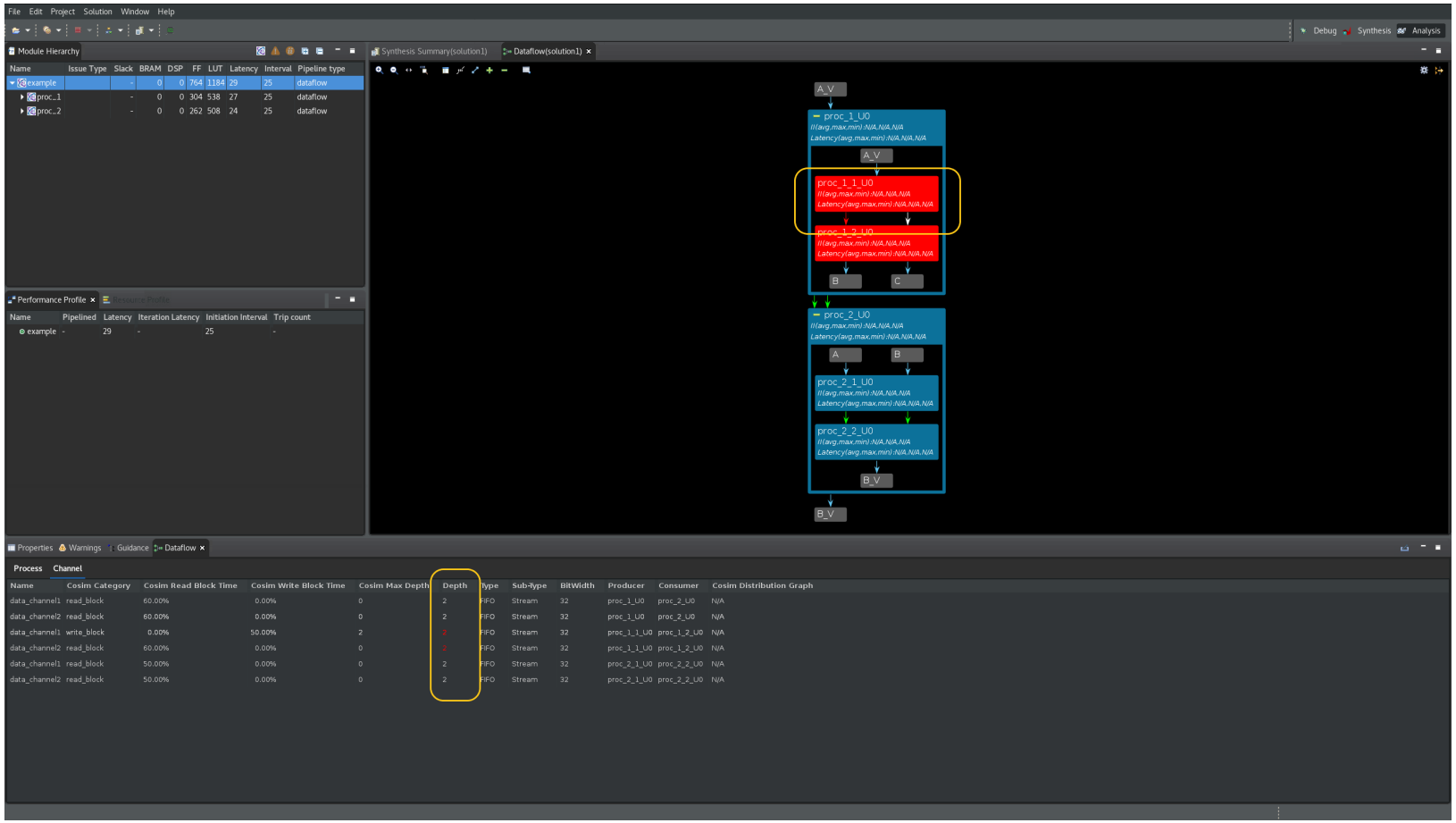
Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

Deadlock Detection and Analysis



Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

Deadlock Detection and Analysis



Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

Deadlock Detection and Analysis

- There are three ways to do this FIFO sizing, and this lab will walk through each one in turn:
  - Manual FIFO sizing
  - Global FIFO sizing
  - Automated FIFO sizing

Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

Manual FIFO sizing

Properties | Warnings | Guidance | Dataflow ×

Process | Channel

Name	Cosim Category	Cosim Read Block Time	Cosim Write Block Time	Cosim Max Depth	Depth	Type	Sub-Type	BitWidth	Producer	Consumer	Cosim Distribution Graph
data_channel1	read_block	60.00%	0.00%	0	2	FIFO	Stream	32	proc_1_U0	proc_2_U0	N/A
data_channel2	read_block	60.00%	0.00%	0	2	FIFO	Stream	32	proc_1_U0	proc_2_U0	N/A
data_channel1	write_block	0.00%	50.00%	2	2	FIFO	Stream	32	proc_1_1_U0	proc_1_2_U0	N/A
data_channel2	read_block	60.00%	0.00%	0	2					c_1_2_U0	N/A
data_channel1	read_block	50.00%	0.00%	0	2					c_2_2_U0	N/A
data_channel2	read_block	50.00%	0.00%	0	2					c_2_2_U0	N/A

Show Source
Modify Depth
Back Annotate The New Depth Into The Design

Modify Depth (on xsjrdevl112) ×

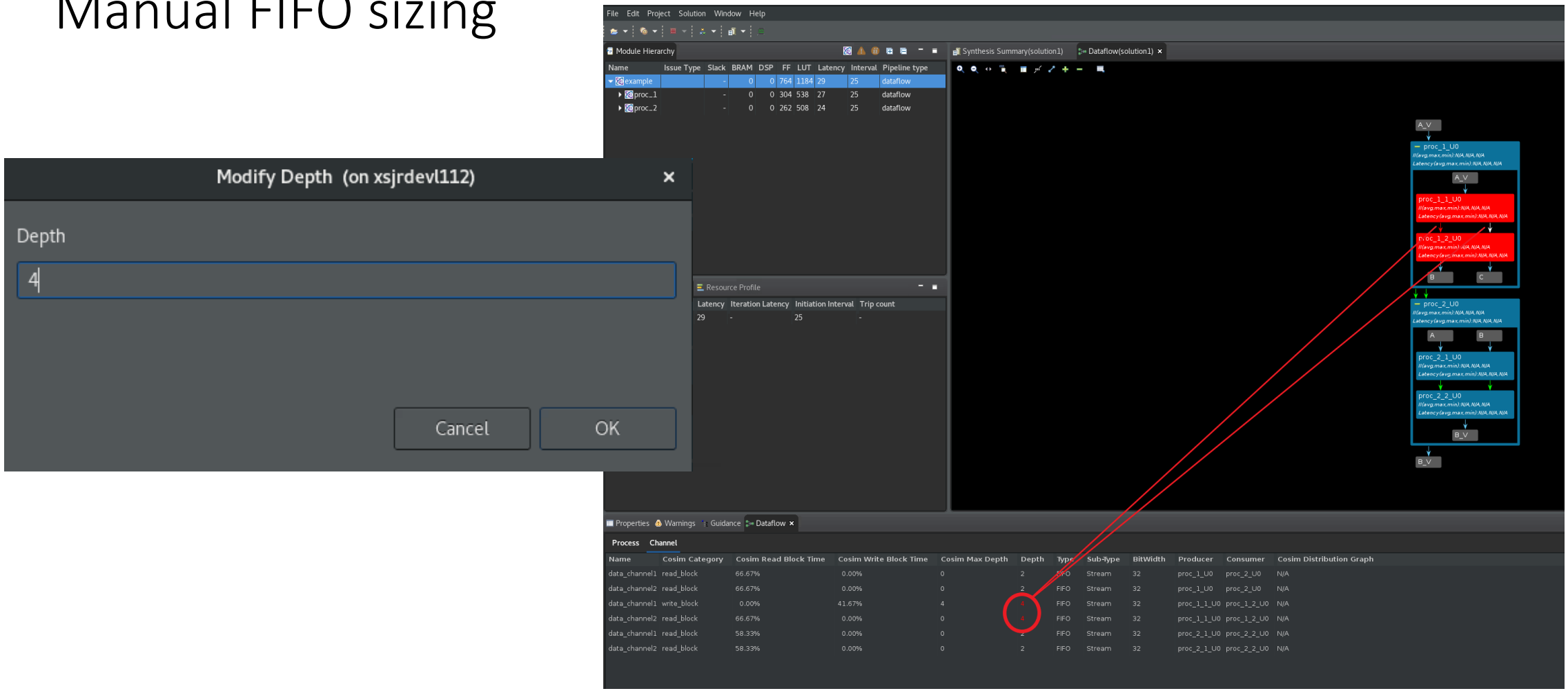
Depth

4

Cancel | OK

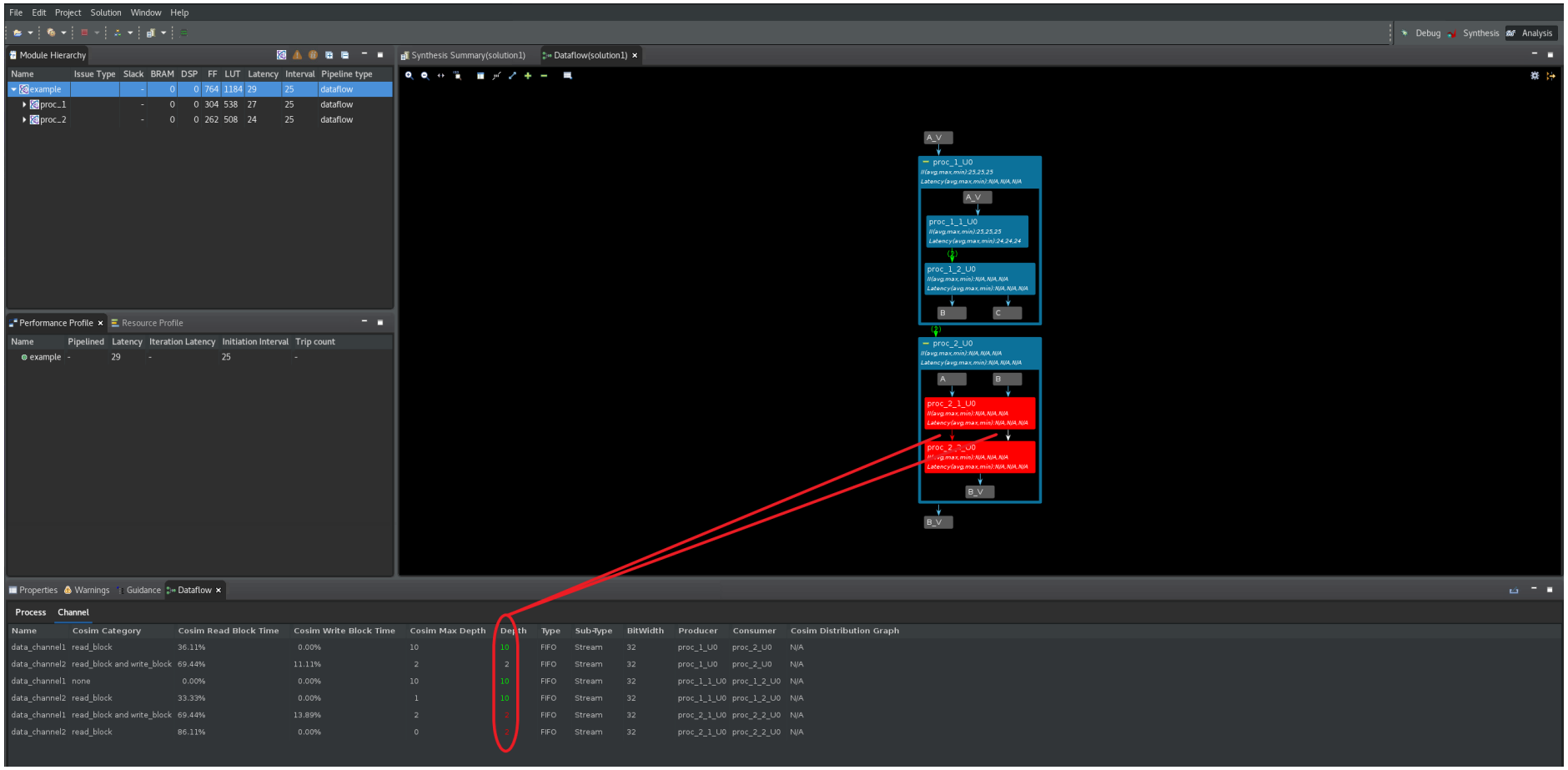
Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

Manual FIFO sizing



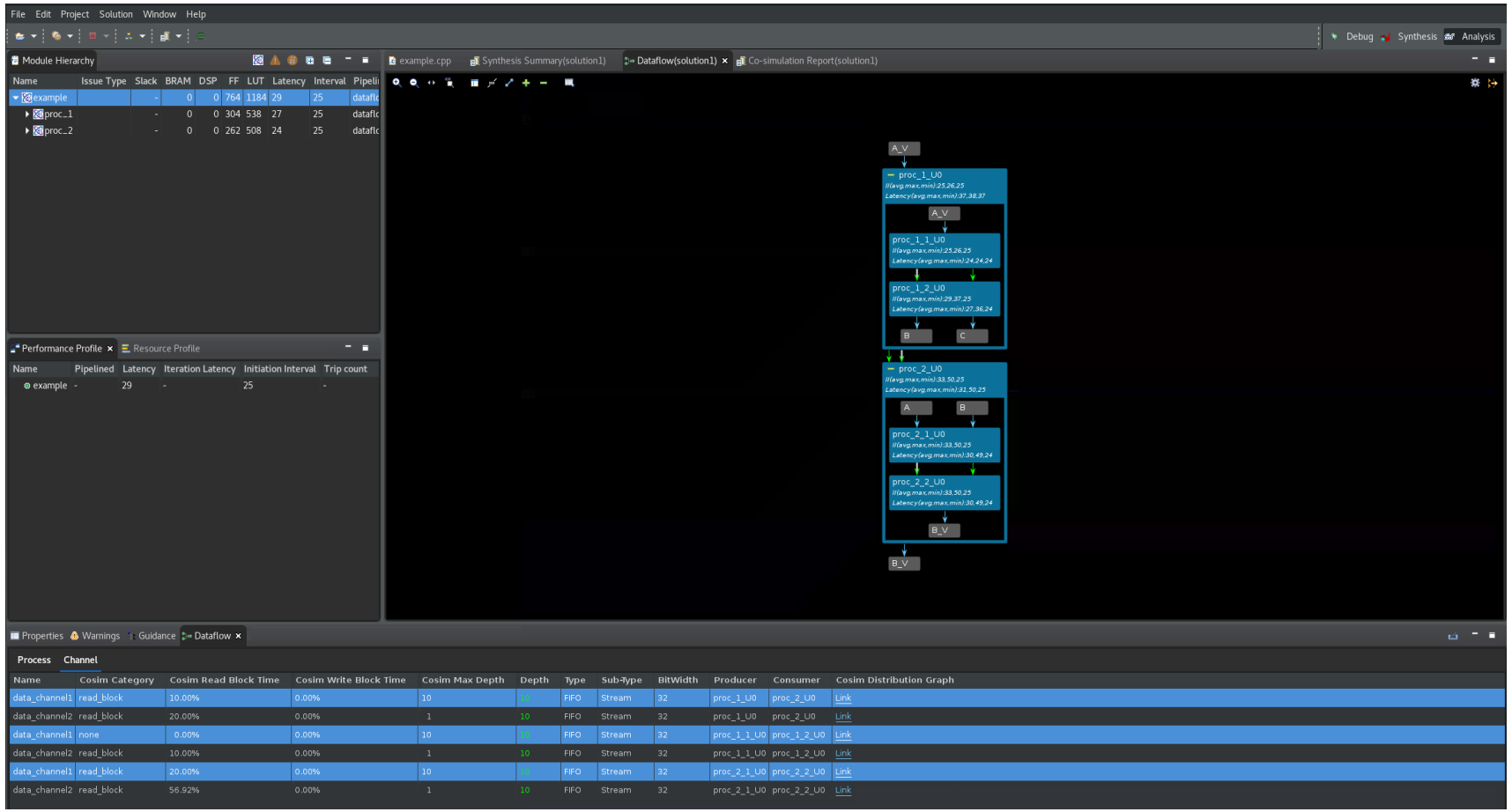
Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

Manual FIFO sizing



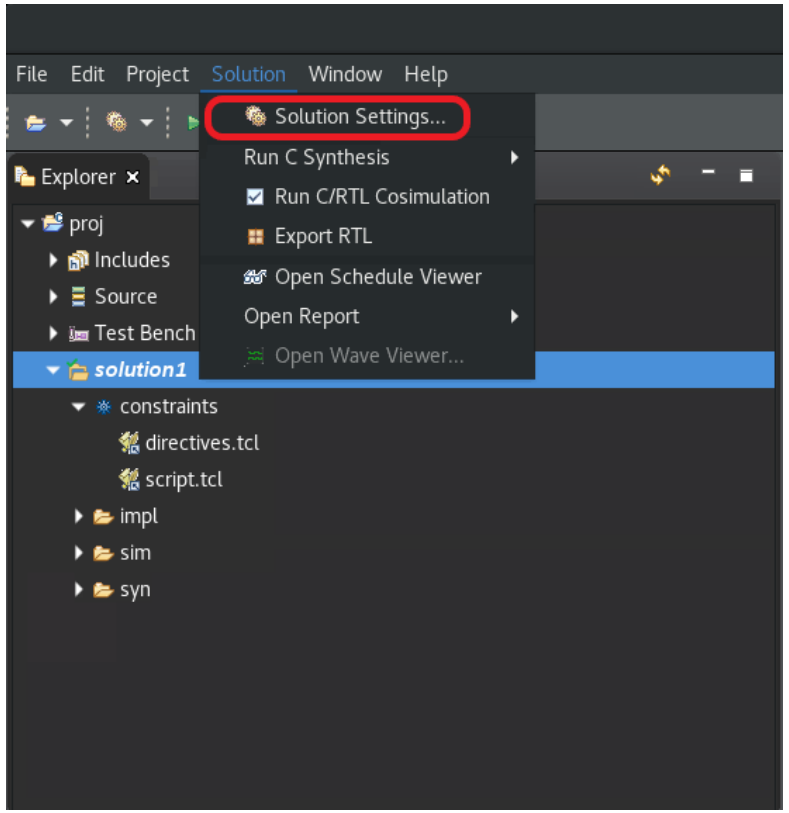
Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

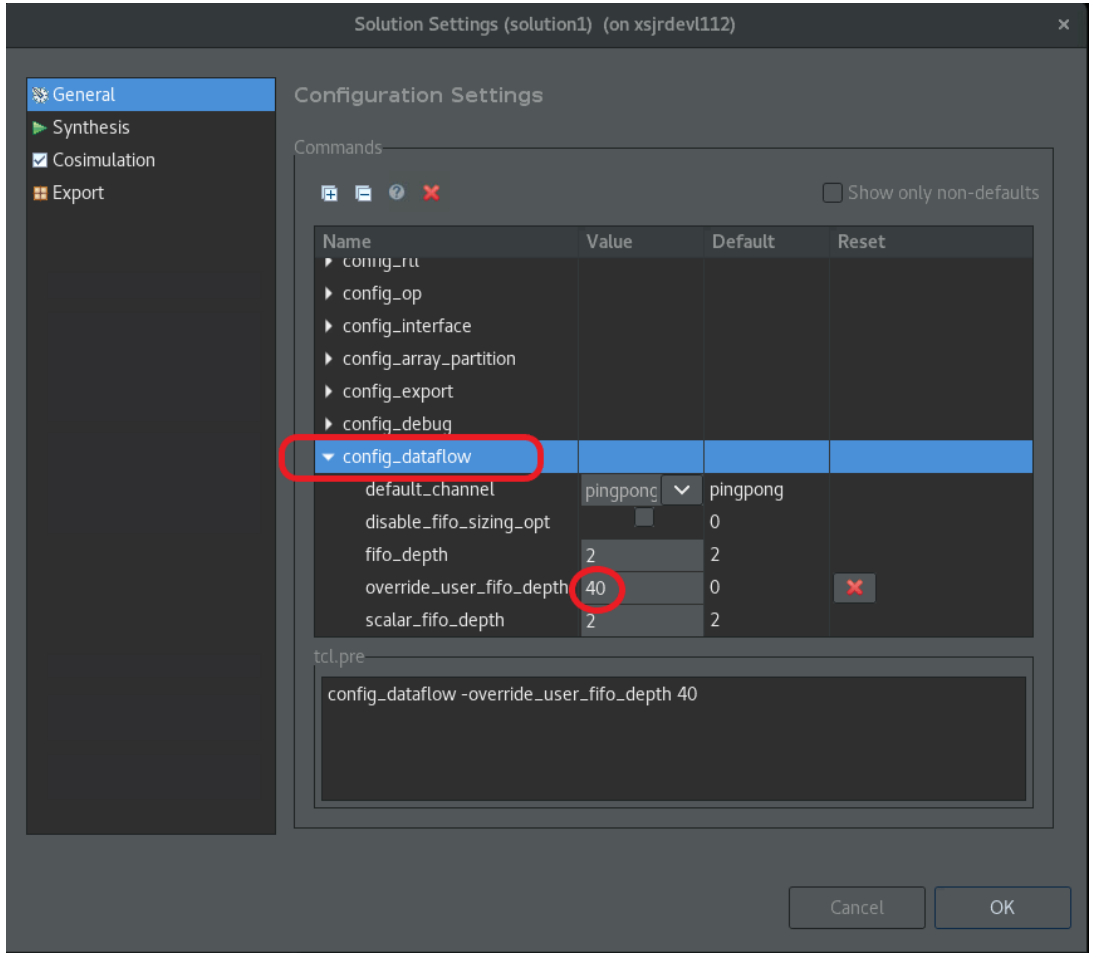
Manual FIFO sizing



Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

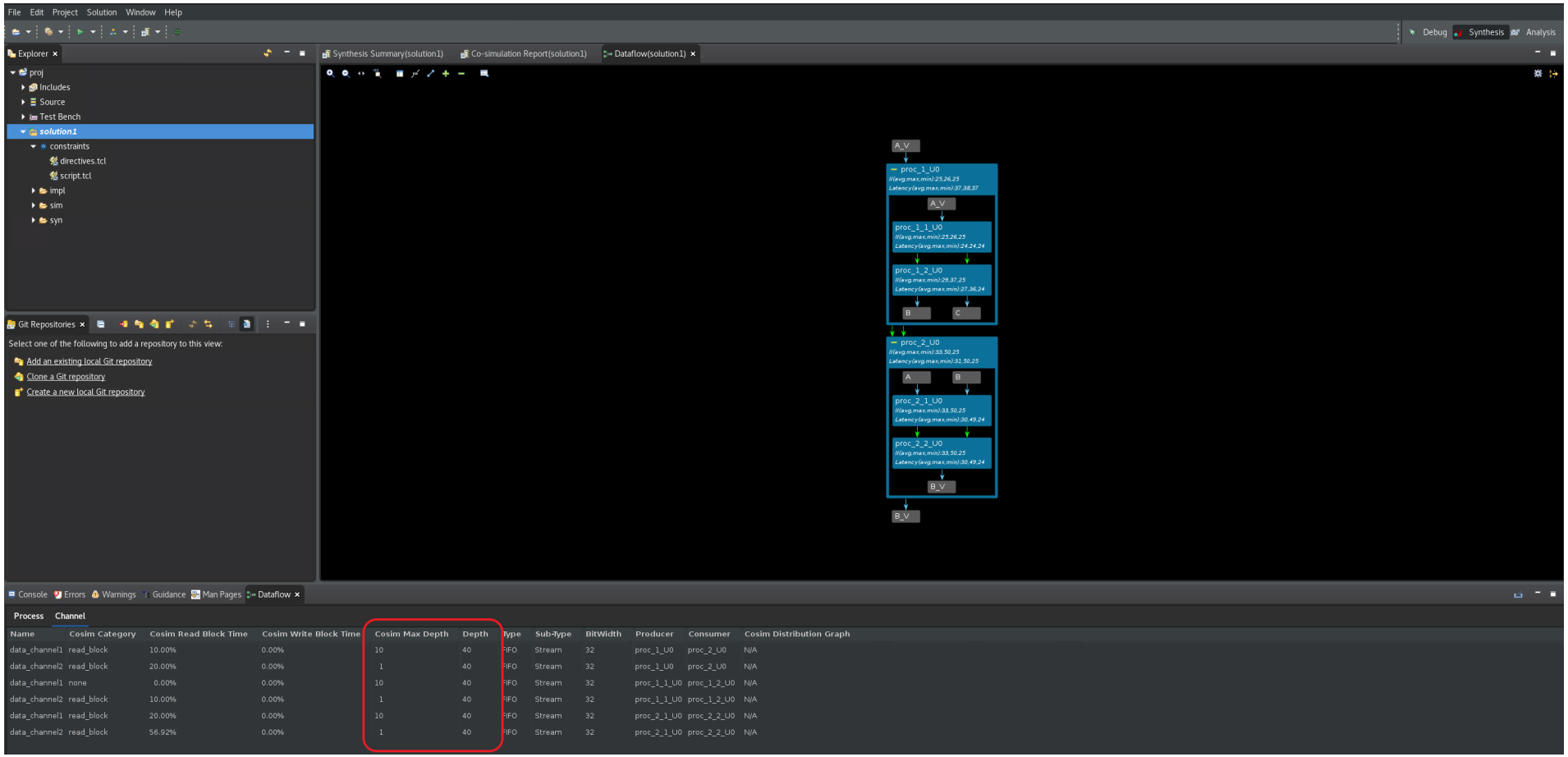
Global FIFO sizing





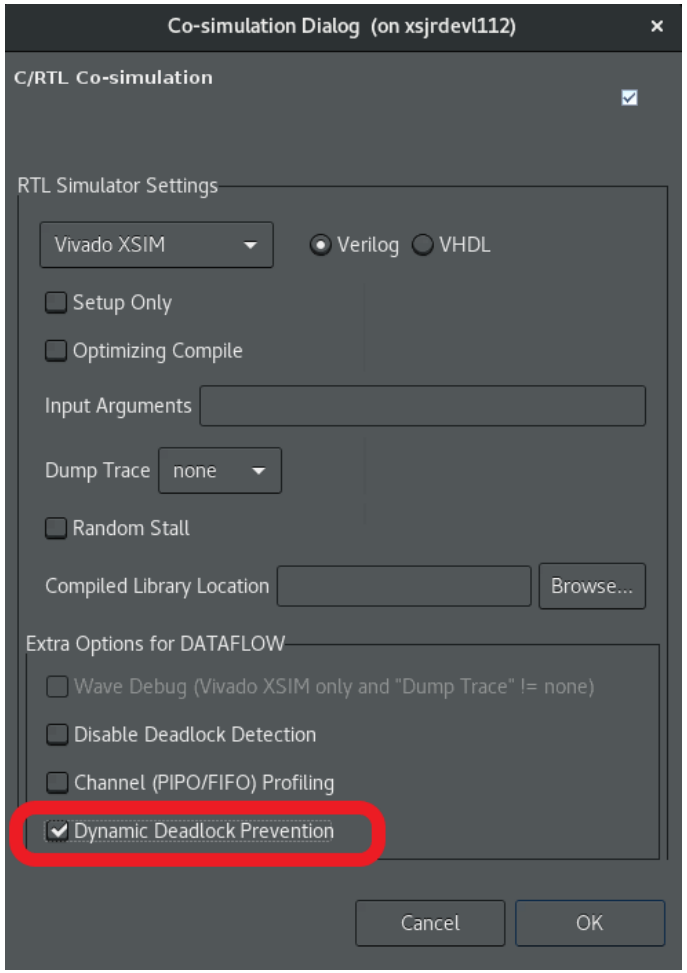
Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

Global FIFO sizing



Second Lab – FIFO Sizing for Performance and avoiding Deadlocks

Automated FIFO sizing



Name	Suggest Depth	Type
data_channel1	11	Stream
data_channel2	2	Stream
data_channel1	11	Stream
data_channel2	2	Stream
data_channel1	11	Stream
data_channel2	2	Stream